



GET TO LINUX

Start now!



OPEN SOURCE COMMUNITY

Text Processing

Let's start?



Agenda

- Grep
- Regex
- Sed
- Awk
- Move files to different folders based on the creation date



grep

The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern. The pattern that is searched in the file is referred to as the regular expression.



grep [options] [pattern] [files]

Option	Function
-i	Match both (upper and lower) case.
-n	Shows the matching line and its number.
-v	Shows all the lines that do not match the searched string.
-c	Displays only the count of matching lines.
-o	Print only the matched parts of a matching line, with each such part on a separate output line.
-w	Match whole word.
-A n	Prints searched line and nlines after the result.
-B n	Prints searched line and n line before the result.
-C n	Prints searched line and n lines after before the result.
-h	Display the matched lines, but do not display the filenames.
-l	Displays list of a filenames only.
-e	Specifies expression with this option. Can use multiple times.
-f	Takes patterns from file, one per line.



Regular Expressions

Regular Expressions are special characters which help search data and matching complex patterns.

Command [Pattern] [File Name]

Symbol	Description
.	Replaces any character.
^	Matches beginning of line.
\$	Matches end of line.
*	Matches zero or more instance of the preceding character.
[]	To match specific characters.
[^]	Exceptions in a character set.



1. Match a single character that is a letter, number, or underscore.
2. Match lines ending with **X**
3. Match the lines ending with '.' (dot)
4. Match lines starting with **Th**
5. Match any word that starting with **o** and ending with **g**



Answers

1. Match a single character that is a letter, number, or underscore. **Ans : `[A-Za-z0-9_]`**
2. Match lines ending with **X** **Ans : `X$`**
3. Match the lines ending with '.' (dot) **Ans : `\.$`**
4. Match lines starting with **Th** **Ans : `^Th`**
5. Match any word that starting with **o** and ending with **g** **Ans : `o[a-z]*g`**



sed

Sed command can perform lots of functions on file like searching, find and replace, insertion or deletion.



The Substitute Command **s**

Syntax : `sed 's/pattern/Replacement_string/'`

s for Substitution changes all occurrences of the regular expression into a new value.

Example:

Changing the word "day" to "night": `sed 's/day/night/' fileName`

By default, the sed command replaces the first occurrence of the pattern in each line and it won't replace the second, third...occurrence in the line.

Find & Replace

Flag	Description	Example
/g	Replacing all the occurrence in a line	<code>sed 's/unix/linux/g' file.txt</code>
/n	Replacing the nth occurrence in a line	<code>sed 's/unix/linux/2' file.txt</code>
/ng	Replacing from nth occurrence to all occurrences in a line	<code>sed 's/unix/linux/3g' file.txt</code>
n s/	Replacing string on a specific line number.	<code>sed '3 s/unix/linux/' file.txt</code>
n,m s/	Replacing string on a range of lines	<code>sed '1,3 s/unix/linux/' file.txt</code>
-n /p	Printing only the replaced lines	<code>sed -n 's/unix/linux/p' file.txt</code>
/I	Ignore case	<code>sed 's/unix/linux/i' file.txt</code>
/pattern/	Replace on a line which matches a pattern.	<code>sed '/linux/ s/unix/centos/' file.txt</code>
/c	Change the entire line with a new line.	

Deletion

Flag	Description	Example
'nd'	To delete a particular line	<code>sed '5d' filename.txt</code>
'\$d'	To delete the last line	<code>sed '\$d' filename.txt</code>
'x,yd'	To delete line from range x to y	<code>sed '3,6d' filename.txt</code>
'nth,\$d'	To delete from nth to last line	<code>sed '12,\$d' filename.txt</code>
'/pattern/d'	To delete pattern matching line	<code>sed '/abc/d' filename.txt</code>



5-Minute Break



awk

Awk is a programming language which allows easy manipulation of structured data. Like common programming language, Awk has variables, conditionals, loops, arithmetic and string operators.

It is mostly used for advanced text processing.



How AWK Works

Records and Fields

Awk can process textual data files and streams. The input data is divided into records and fields. Awk operates on one record at a time until the end of the input is reached.

Records are separated by a character called the **record separator**. The default record separator is the newline character, which means that each line in the text data is a record.

Records consist of fields which are separated by the **field separator**. By default, the value of the field separator is any number of space or tab characters.

How AWK Works

The fields in each record are referenced by the dollar sign (\$) followed by field number, beginning with 1. The first field is represented with \$1, the second with \$2, and so on. The last field can also be referenced with the special variable \$NF. The entire record can be referenced with \$0.

```
salma@fedora:~  
[~]$ df -h ~/HDD/  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/sda1       932G  539G  393G   58% /home/salma/HDD
```

\$6 \$7 --> (\$NF)

\$1 \$2 \$3 \$4 \$5 \$6 --> Fields

\$0 --> Record

```
salma@fedora:~/awk
[awk]$ cat info
fristName      lastName      age      city      ID
Salma          Ayman         20       Cairo     100
Mohamed        El-Sayed     15       Berlin    300
Lolo           Alassal      55       Aswan     900
Jena           Max          65       London    200
[awk]$ awk '{ print $0 }' info
fristName      lastName      age      city      ID
Salma          Ayman         20       Cairo     100
Mohamed        El-Sayed     15       Berlin    300
Lolo           Alassal      55       Aswan     900
Jena           Max          65       London    200
[awk]$ awk '{print $1 $2}' info
fristNamelastName
SalmaAyman
MohamedEl-Sayed
LoloAlassal
JenaMax
[awk]$ awk '{print $1,$2}' info
fristName lastName
Salma Ayman
Mohamed El-Sayed
Lolo Alassal
Jena Max
[awk]$ awk '{print $NF}' info
ID
100
300
900
200
[awk]$
```



**Move files to different
folders based on the
creation date**

Open Source Community

How to Contact Us



[Discord](#)



[Facebook](#)



[Linkedin](#)



[Instagram](#)



[Twitter](#)



Thank you

